

Computación Distribuida Asíncrona en redes heterogéneas usando una máquina virtual Java

María Isabel García Arenas

Depto. Informática, Universidad de Jaén
Paraje Las Lagunillas, s/n
Jaén, 23071 mgarenas@ujaen.es

Resumen

Este artículo presenta un resumen global de la tesis titulada *Computación Evolutiva Distribuida Asíncrona en redes heterogéneas usando una máquina virtual Java*. El trabajo incluye una introducción general a la Computación Evolutiva en redes heterogéneas, una descripción básica de los aspectos más técnicos del diseño de la herramienta y del tipo de paralelismo que utiliza, un resumen de los resultados obtenidos y una descripción de la aportación original de la tesis.

Palabras clave: Computación Evolutiva, Paralelismo Asíncrono, Distribución de procesos.

1. Introducción

Esta tesis está motivada por la necesidad de herramientas de Computación Evolutiva [14, 9] (CE) que faciliten la creación de experimentos para aprovechar la potencia disponible en máquinas no locales conectadas a una red como puede ser Internet. Esta herramienta, denominada JEO (*Objetos Evolutivos en Java*), es lo suficientemente flexible, potente y fácil de usar como para construir cualquier experimento de forma distribuida. Otra de las características más destacables es que permite la creación de experimentos escalables posibilitando así centrar la investigación en el propio experimento y despreocuparse de otros aspectos que pueden no estar tan ligados a los algoritmos. La escalabilidad que adquieren los experimentos construidos con JEO está avalada por un soporte software que permite la construcción de una red punto a punto que implementa un modelo de computación paralela basado en el paso de mensajes.

JEO da soporte a la creación de experimentos de los paradigmas tradicionales como son los algorit-

mos genéticos, las estrategias evolutivas, la programación evolutiva o la programación genética. También propone un nuevo modelo general de computación evolutiva basado en una economía virtual que ayuda a la evolución de soluciones. JEO está totalmente integrada en un proyecto denominado DREAM (Máquina Distribuida de Recursos para Algoritmos Evolutivos) [11] que permite crear redes punto a punto [8, 15] de nodos de ejecución para los experimentos que se desarrollan.

Para probar la utilidad de la herramienta se presentan varios problemas clásicos en Computación Evolutiva y algunos problemas que introducen innovaciones en este campo, dotando a JEO de una potencia que otras herramientas no poseen.

2. Descripción de JEO

Se trata de una herramienta que permite a su usuario aprovechar los recursos computacionales que proporciona una red de máquinas heterogéneas y que permite generar experimentos de

CE que aprovechan estos recursos de forma paralela [5, 2, 1]. Las características más destacables de su diseño son la flexibilidad para la generación de experimentos de CE de cualquiera de los paradigmas de computación que la forman, la robustez en la ejecución de dichos experimentos, la independencia de la plataforma de ejecución, permitiendo utilizar redes de computadoras totalmente heterogéneas y la abstracción de la capa de red.

El esquema de computación paralela que utiliza es el llamado *paso de mensajes* basado en la creación de primitivas que permiten la comunicación entre el conjunto de procesadores que componen la máquina paralela de cómputo. Este modelo el que más se ajusta para aprovechar la potencia de procesamiento de una red de computadores no necesariamente homogénea y sus ventajas [12] lo hacen el modelo más adecuado para este trabajo.

Queda ahora determinar la granularidad de los experimentos a generar. Existen dos grandes grupos extremos, los algoritmos paralelos de grano fino [10] y los algoritmos paralelos de grano grueso [7].

En los algoritmos de grano fino, se crean poblaciones de pocos individuos que se distribuyen entre los diferentes procesadores de que se disponga manteniendo un patrón entre las poblaciones que puede ser por ejemplo un anillo, una matriz bidimensional o tridimensional que establecen entre ellas comunicaciones siguiendo una función de vecindad. Hoy en día este modelo está dejando paso a los algoritmos de grano grueso, puesto que estas implementaciones de grano fino necesitan de hardware más específico, normalmente un conjunto de procesadores homogéneo con memoria compartida, para alcanzar una ganancia en velocidad aceptable.

En los algoritmos de grano grueso, la idea principal de este modelo es que poblaciones aisladas que compiten son más efectivas en la búsqueda de una solución que una sola población global. Las poblaciones, o islas, pueden intercambiar individuos cada cierto tiempo y este intercambio se denomina migración. Las migraciones de individuos pueden ajustarse a varios patrones y la principal función de un individuo que emigra es la de aportar diversidad a la población o isla a la que llega.

JEO está diseñada para aprovechar la potencia de cómputo que aporta una red heterogénea de computadoras [3] y su tolerancia a fallos debe ser máxima, puesto que no se trata de una red dedicada a este tipo de procesamiento. Por esta

razón, los algoritmos generados son algoritmos de grano grueso en los que el patrón de migraciones entre las diferentes poblaciones puede ser totalmente dinámico y asíncrono, permitiéndole que la ejecución sea totalmente independiente de la capa física que lo está ejecutando.

El soporte software necesario para las diferentes comunicaciones entre poblaciones, se implementa en una capa inferior a JEO que le proporciona los servicios básicos de envío y recepción de mensajes denominada DRM [13].

2.1. Diseño de Clases

Todos los objetos que forman parte de la herramienta están diseñados bajo un interfaz de funcionamiento. Esta característica hace que la ampliación de la herramienta sea rápida y sencilla. El esquema de relación entre las clases permite crear experimentos no sólo evolutivos sino también coevolutivos. La coevolución de especies está basada en la posibilidad de crear espacios de evolución totalmente independientes entre ellos en los que las variables que lo conducen en la búsqueda de soluciones sean totalmente diferentes y permitan comparar en un mismo experimento, por ejemplo, qué característica afecta más directamente a una solución o qué población ha evolucionado mejor bajo un determinado entorno. La coevolución [4], un tema en auge en los últimos años, y JEO incorpora esta posibilidad dándole dos posibles enfoques, una coevolución competitiva o una coevolución cooperativa.

3. Experimentos

Se presentan varios experimentos de varios ámbitos diferentes en CE, como son la programación genética y los algoritmos genéticos con codificación real. Se realizan varios experimentos orientados a probar la ganancia obtenida en cada conjunto de experimentos utilizando un número de computadoras variable y una complejidad del problema variable.

Se presentan la ganancia obtenida para dos problemas, la optimización de la función de Ackley utilizando un algoritmo genético con codificación real (figura 1) y un problema de regresión clásica (figura 2) realizado con programación genética [6].

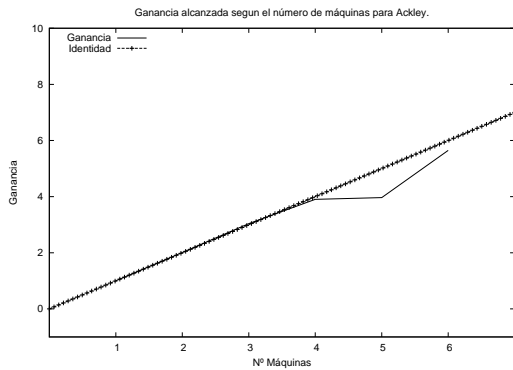


Figura 1

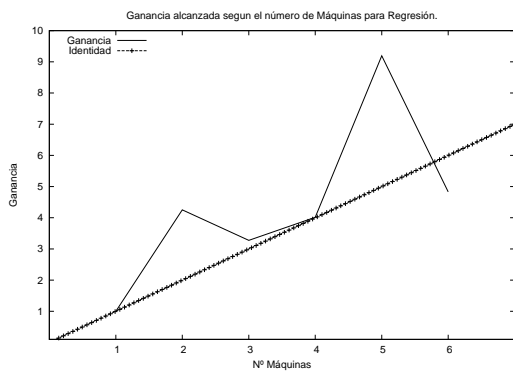


Figura 2

población, eliminando malas soluciones a lo largo de la evolución o mecanismos de regeneración de poblaciones utilizando individuos con un poder económico elevado que se pueden permitir generar más individuos a partir de ellos mismos.

Este sistema económico también se podría utilizar para controlar cuantos recursos de cómputo puede utilizar una población en un determinado computador, permitiendo a los usuarios que unan sus recursos a la ejecución de un experimento o decidir cuántos recursos pone a disposición de la ejecución.

Los resultados obtenidos para experimentos en los que se utiliza esta característica se pueden ver en la figura 3.

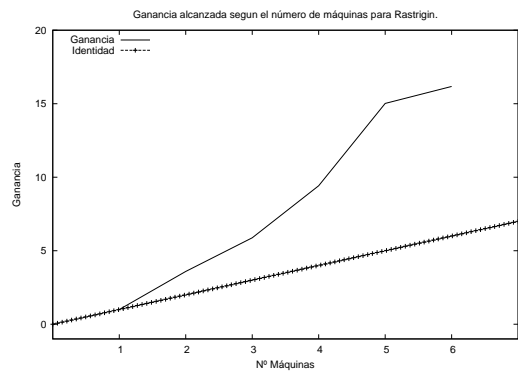


Figura 3

4. Aportaciones Originales

La aportación original de JEO es que plantea un sistema económico para los individuos que pertenecen a una población. Este sistema económico puede permitir que cada población establezca las reglas para obtener, por parte de un individuo un beneficio como puede ser emigrar a otra población o generar un nuevo individuo con una operación genética (cruce o mutación por ejemplo).

El sistema económico establece clases en los individuos de una población. Clases que representan buenas soluciones, que obtienen mucho beneficio de sus operaciones, o que representan malas soluciones, es decir, que no han obtenido beneficios de sus operaciones.

Este sistema económico introduce un grado más de la realidad de una población en un algoritmo evolutivo, dotando a los individuos de un estamento económico que puede ser utilizado por ejemplo para establecer mecanismos de control de

5. Agradecimientos

Esta tesis ha sido desarrollada en el departamento de Arquitectura y Tecnología de los computadores de la Universidad de Granada, bajo la atenta dirección de los doctores Pedro A. Castillo Valdivieso y Juan Julián Merelo Guervós.

Referencias

[1] M. G. Arenas, B. Dolin, and J.J. Merelo. Opposites attract: Complementary phenotype selection for crossover in genetic programming. In *Proceedings of the Seventh Conference on Parallel Problem Solving from Nature*, volume 2439 of *Lecture Notes in Computer Science*, pages 142–152, Granada, Spain, September 7-11 2002. Springer-Verlag.

- [2] M. G. Arenas, Brad Dolin, Juan Julián Merelo, Pedro Angel Castillo, Ignacio Fernández De Viana, and Marc Schoenauer. JEO: Java evolving objects. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honnavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, page 991, New York, 9-13 July 2002. Morgan Kaufmann Publishers.
- [3] M.G. Arenas, P. A. Castillo, G. Romero, and J.J. Merelo. Distribución de información en algoritmos evolutivos p2p. In Universidad de Oviedo, editor, *Actas del II Congreso español sobre Metaheurísticas, Algoritmos Evolutivos y bioinspirados*, pages 85–92, Febrero 2003.
- [4] M.G. Arenas, P.A. Castillo, G. Romero, V. Rivas, F. Rateb, and J.J. Merelo. Co-evolving multilayer perceptrons along training sets. *Advances in Soft Computing*, September 2005.
- [5] M.G. Arenas, L. Foucart, J. J. Merelo, and P. A. Castillo. JEO: A Framework for Evolving Objects in Java. In Universidad Politécnica de Valencia, editor, *In XII Jornadas de Paralelismo*, pages 185–191, 46071 Valencia, September 2001. REPROVAL, S.L.
- [6] M.G Arenas, L. Foucart, M. Shoenauer, and J.J. Merelo. Computación evolutiva en Java: JEO. In Universidad de Mérida, editor, *Actas del I Congreso Español de Algoritmos Evolutivos y Bioinspirados*, pages 46–53, Mérida, Badajoz, February 2002.
- [7] E. Cantú-Paz. A survey of parallel genetic algorithms. Technical Report 97003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.
- [8] Peer To Peer Communications. Computers books by computer experts. Disponible en <http://www.peer-to-peer.com>.
- [9] Oscar Cordón. *Una metodología para el diseño automático de sistemas basados en reglas difusas mediante algoritmos evolutivos*. Tesis, Universidad de Granada, Granada, Spain, Julio 1997.
- [10] El ghazali Talbi and Pierre Bessiere. A parallel genetic algorithm for the graph partitioning problem. In *ACM Int. Conf. on Supercomputing ICS91*, Cologne, Germany, July 1991.
- [11] Li Gong. Peer-to-peer networks in action. *IEEE Internet Computing online*, 2002.
- [12] Parallel Research Group. Advantages of the message-passing model. Available in <http://prg.cpe.ku.ac.th/~thvo/mpith/thesis/mpith-proposal/-node13.html>, 2002.
- [13] Mark Jelasity, Mike Preub, Maarten van Steen, and Ben Paechter. Maintaining connectivity in a scalable and robust distributed environment. In *2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002)*, pages 389–394, May 2002.
- [14] Heitkötter Jörg and Beasley David. Hitchhiker’s guide to evolutionary computation. Available in <http://www.faqs.org/faqs/-ai-faq/genetic/part1/preamble.html>, Dec 2001.
- [15] Nelson Minar, Marc Hedlund, Clay Shirky, Tim O’Reilly, Dan Bricklin, David Anderson, Jeremie Miller, Adam Langley, Gene Kan, Alan Brown, Marc Waldman, Lorrie Cranor, Aviel Rubin, Roger Dingledine, Michael Freedman, David Molnar, Rael Dornfest, Dan Brickley, Theodore Hong, Richard Lethin, Jon Udell, Nimisha Asthagiri, Walter Tuvell, Brandon Wiley, and Avinash Chugh. *Peer-to-Peer. Harnessing the Power of Disruptive Technologies*. O’Reilly, 2001.